Foundation of Cryptography (0368-4162-01), Lecture 7

MACs and Signatures

Iftach Haitner, Tel Aviv University

December 27, 2011

Section 1

Message Authentication Code (MAC)

Message Authentication Code (MAC)

Goal: message authentication.

Message Authentication Code (MAC)

Goal: message authentication.

Definition 1 (MAC)

A MAC is a tuple of PPT's (Gen, Mac, Vrfy) such that

1. $\mathsf{Gen}(1^n)$ outputs a key $k \in \{0,1\}^*$
2. $\mathsf{Mac}(k, m)$ outputs a “tag" t
3. $\mathsf{Vrfy}(k, m, t)$ output 1 (YES) or 0 (NO)

Message Authentication Code (MAC)

Goal: message authentication.

Definition 1 (MAC)

A MAC is a tuple of PPT's (Gen, Mac, Vrfy) such that

1. $\mathrm{Gen}(1^n)$ outputs a key $k \in \{0,1\}^*$
2. $\mathrm{Mac}(k, m)$ outputs a "tag" t
3. $\mathrm{Vrfy}(k, m, t)$ output 1 (YES) or 0 (NO)

We require

Consistency: $\mathrm{Vrfy}(k, m, t) = 1$ for any $k \in \mathrm{Supp}(\mathrm{Gen}(1^n))$, $m \in \{0,1\}^n$ and $t = \mathrm{Mac}(k, m)$

Unforgability: No PPT wins the MAC game with respect to (Gen, Mac, Vrfy)

Definition 2 (MAC game)

Let $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$, if the following is not negligible:

$$(m, t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n, m, t) = 1$$
$$\wedge\ \mathrm{Mac}(K_n, \cdot) \text{ was not asked on } m$$

Definition 2 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m, t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n, m, t) = 1$$
$$\wedge\, \mathrm{Mac}(K_n, \cdot) \text{ was not asked on } m$$

- “Private key" definition

Definition 2 (MAC game)

Let $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to $(\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$, if the following is not negligible:

$$(m, t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n, m, t) = 1 \\ \wedge \mathrm{Mac}(K_n, \cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?

Definition 2 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \text{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m, t) \leftarrow A^{\text{Mac}(K_n,\cdot),\text{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \text{Vrfy}(K_n, m, t) = 1$$
$$\wedge\, \text{Mac}(K_n, \cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?
- Definition too strong? Any message? Use of Verifier?

Definition 2 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m,t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n,m,t) = 1$$
$$\wedge\ \mathrm{Mac}(K_n,\cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?
- Definition too strong? Any message? Use of Verifier?
- “Reply attacks"

Definition 2 (MAC game)

Let (Gen, Mac, Vrfy) be a MAC and let $K_n = \mathrm{Gen}(1^n)$. An oracle-aided algorithm A wins the MAC game with respect to (Gen, Mac, Vrfy), if the following is not negligible:

$$(m,t) \leftarrow A^{\mathrm{Mac}(K_n,\cdot),\mathrm{Vrfy}(K_n,\cdot,\cdot)}(1^n) \wedge \mathrm{Vrfy}(K_n,m,t) = 1$$
$$\wedge\ \mathrm{Mac}(K_n,\cdot) \text{ was not asked on } m$$

- “Private key" definition
- Variable length messages?
- Definition too strong? Any message? Use of Verifier?
- “Reply attacks"

Definition 3 (ℓ-time MAC)

Same as in Definition 1, but security is only required against ℓ-query adversaries.

constructions

Construction 4 (One-time MAC)

$\mathrm{Gen}(1^n) = U_n$, $\mathrm{Mac}(k, m) = k \oplus m$ and $\mathrm{Vrfy}(k, m, t) = 1$ iff $t = k \oplus m$

constructions

Construction 4 (One-time MAC)

$\mathrm{Gen}(1^n) = U_n$, $\mathrm{Mac}(k, m) = k \oplus m$ and $\mathrm{Vrfy}(k, m, t) = 1$ iff $t = k \oplus m$

Construction 5 ($\ell \in$ poly-time MAC, Stateful)

Use ℓ random strings of length n

constructions

Construction 4 (One-time MAC)

$\mathrm{Gen}(1^n) = U_n$, $\mathrm{Mac}(k, m) = k \oplus m$ and $\mathrm{Vrfy}(k, m, t) = 1$ iff $t = k \oplus m$

Construction 5 ($\ell \in$ poly-time MAC, Stateful)

Use ℓ random strings of length n

Construction 6 ($\ell \in$ poly-time MAC)

$\mathrm{Gen}(1^n)$ return a random member in $\mathcal{H}_n$, where $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$ is an efficient family of ℓ-wise independent hash functions.[a]
Let $\mathrm{Mac}(k, m) = k(m)$, and $\mathrm{Vrfy}(k, m, t) = 1$ iff $t = k(m)$.

[a] For any distinct $x_1, \ldots, x_\ell \in \{0,1\}^n$ and $y_1, \ldots, y_\ell \in \{0,1\}^n$, $\Pr h \leftarrow \mathcal{H}_n[h(x_1) = y_1 \wedge \cdots \wedge h(x_\ell) = y_\ell] = 2^{-tn}$.

PRF-based MAC

Construction 7 (PRF-based MAC)

Same as Construction 6, but uses a family of length preserving function $\mathcal{F}$ instead of $\mathcal{H}$.

Claim 8

Assuming that $\mathcal{F}$ is a PRF, then Construction 7 is a (poly-time) MAC.

Proof:

PRF-based MAC

Construction 7 (PRF-based MAC)

Same as Construction 6, but uses a family of length preserving function $\mathcal{F}$ instead of $\mathcal{H}$.

Claim 8

Assuming that $\mathcal{F}$ is a PRF, then Construction 7 is a (poly-time) MAC.

Proof: Easy to prove if $\mathcal{F}$ is a family of random functions. Hence, also holds in case $\mathcal{F}$ is a PRF. $\square$